MPRA
Munich Personal RePEc Archive

Pricing Early-Exercise and Discrete Barrier Options by Fourier-Cosine Series Expansions

Fang Fang and Kees Oosterlee

Delft University of Technology

June 2008

Online at http://mpra.ub.uni-muenchen.de/9248/
MPRA Paper No. 9248, posted 29. June 2008 02:42 UTC

Pricing Early-Exercise and Discrete Barrier Options by Fourier-Cosine Series Expansions

F. Fang[*] C.W. Oosterlee[†]

June 17, 2008

Abstract

We present a pricing method based on Fourier-cosine expansions for early-exercise and discretely-monitored barrier options. The method works well for exponential Lévy asset price models. The error convergence is exponential for processes characterized by very smooth ($C^\infty[a,b] \in \mathbb{R}$) transitional probability density functions. The computational complexity is $O((M-1)N \log N)$ with N a (small) number of terms from the series expansion, and M, the number of early-exercise/monitoring dates. This paper is the follow-up of [22] in which we presented the impressive performance of the Fourier-cosine series method for European options.

1 Introduction

Within stock option pricing applications, interesting numerical mathematics questions can be found in product pricing and in calibration. Whereas the former topic requires especially robust numerical techniques, the latter also relies on efficiency and speed of computation.

Numerical integration methods, based on a transformation to the Fourier domain (the so-called *transform methods*), are traditionally very efficient, due to the availability of the Fast Fourier Transform (FFT) [13, 35], for the pricing of basic European products, and thus for calibration purposes. These methods can readily be applied to solving problems under various asset price dynamics, for which the characteristic function (i.e., the Fourier transform of the probability density function) is available. This is the case for models from the class of regular affine processes of [19], which also includes the exponentially affine jump-diffusion class of [18], and, in particular, the exponentially Lévy models.

Recently, transform methods have been generalized to solving somewhat more complicated option contracts, like Bermudan, American or barrier options, see, for example, [32, 21, 3, 4, 29, 41, 17, 40]. These exotic options, still with basic features, are used in the financial industry as building blocks for more complicated products. A natural aim for the near future with these transform

*Delft University of Technology, Delft Institute of Applied Mathematics, Delft, the Netherlands, email: f.fang@ewi.tudelft.nl

†CWI – Centrum Wiskunde & Informatica, Amsterdam, the Netherlands, email: c.w.oosterlee@cwi.nl, and Delft University of Technology, Delft Institute of Applied Mathematics.

methods is to calibrate to these exotic products and to price the huge portfolios (at the end of a trading day) very fast.

Next to FFT-based methods, new techniques based on the Fast Gauss or the Hilbert Transform have been introduced for this purpose [9, 10, 23]. In this paper we will also generalize a transform method to pricing Bermudan, American and discretely-monitored barrier options. It is the method based on Fourier-cosine series expansions, called the COS method, introduced by us in [22], where we showed that it was highly efficient for pricing European options. The underlying idea is to replace the transitional probability density function by its Fourier-cosine series expansion, which has an elegant relation to the conditional characteristic function. For many underlying asset price models, the method is remarkably fast and the density function can be recovered easily. Since a whole *function of option values* is obtained, the Greeks can be computed at almost no additional computational cost. Here, we will show that the COS method can also price the early-exercise and barrier options with exponential convergence under various Lévy models.

The methods are, for these option contracts in competition with the methods that require the solution of discrete partial (integro-) differential equation-based operators (PIDE) [44, 11]. PIDE-based methods are traditionally used since early-exercise and the exotic features can often be interpreted as special payoffs or boundary conditions. Generally speaking, however, the computational process with PIDE is rather expensive, especially for the infinite activity Lévy processes we are interested in, because they give rise to an integral in the PIDE with a weakly singular kernel [2, 27, 43].

We will therefore compare our results with other highly efficient transform methods, i.e., with the Convolution (CONV) method [32], based on the FFT, which is one of the state-of-the-art methods for pricing Bermudan and American options. Its computational complexity for pricing a Bermudan option with M exercise dates is $O((M-1)N\log_2(N))$, where N denotes the number of grid points used for numerical integration. Quadrature rule based techniques are, however, not of the highest efficiency when solving Fourier transformed integrals. As these integrands are highly oscillatory, a relatively fine grid has to be used for satisfactory accuracy with the FFT. The COS method presented here requires a substantially smaller value of N.

Especially for barrier options, another highly efficient alternative method from [23] is based on the Hilbert transform. Its error convergence is exponential for models with rapidly decaying characteristic functions, also with a computational complexity of $O((M-1)N\log_2 N)$ for a barrier option with M monitoring dates. This method is, however, not applicable for Bermudan options.

The paper is organized as follows. In Section 2 the COS method for pricing Bermudan and barrier options is presented. The handling of the discretely monitored barrier options is discussed in particular in Subsection 2.4. Error analysis is performed in Section 3. Numerical results are presented in Section 4, where we focus on option pricing under exponential Lévy processes, in particular under the CGMY [12] and the Normal Inverse Gaussian [5] processes.

2 Pricing Bermudan and Barrier Options

A Bermudan option can be exercised at pre-specified dates before maturity. The holder receives the exercise payoff when he/she exercises the option. Between two consecutive exercise dates the valuation process can be regarded as that for a European option, priced with the help of the risk-neutral valuation formula. Let t_0 denote the initial time and $\mathcal{T}\{t_1, \cdots, t_M\}$ be the collection of all exercise dates with $\Delta t := (t_m - t_{m-1})$, $t_0 < t_1 < \cdots < t_M = T$. The pricing formula for a Bermudan option with M exercise dates then reads, for $m = M, M-1, \ldots, 2$:

$$\begin{cases} c(x, t_{m-1}) &= e^{-r\Delta t} \int_{\mathbb{R}} v(y, t_m) f(y|x) dy, \\ v(x, t_{m-1}) &= \max\left(g(x, t_{m-1}), c(x, t_{m-1})\right), \end{cases} \tag{1}$$

followed by

$$v(x, t_0) = e^{-r\Delta t} \int_{\mathbb{R}} v(y, t_1) f(y|x) dy. \tag{2}$$

Here x and y are state variables, defined as the logarithm of the ratio of the asset price S_t over the strike price K,

$$x := \ln(S(t_{m-1})/K) \quad \text{and} \quad y := \ln(S(t_m)/K),$$

$v(x,t)$, $c(x,t)$ and $g(x,t)$ are the option value, the continuation value and the payoff at time t, respectively. Note that for vanilla options, $g(x,t)$ equals $v(x,T)$, with

$$v(x, T) = [\alpha K(e^x - 1)]^+, \quad \alpha = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put.} \end{cases}$$

The probability density function of y given x under a risk-neutral measure is denoted by $f(y|x)$ in (2), and r is the (deterministic) risk-neutral interest rate.

Equations (1), (2) can be efficiently evaluated by the COS method in [22], provided that the Fourier-cosine series coefficients of $v(y, t_m)$ are known.

2.1 The COS Method

The COS method is based on the insight that the Fourier-cosine series coefficients of $f(y|x)$ are closely related to its characteristic function.

Since the density function, $f(y|x)$, decays to zero rapidly as $y \to \pm\infty$, we can truncate the infinite integration range in the risk-neutral valuation formula without loosing significant accuracy. Suppose that we have, with $[a,b] \subset \mathbb{R}$,

$$\int_{\mathbb{R}\setminus[a,b]} f(y|x) dy < \text{ TOL}, \tag{3}$$

for some given tolerance, TOL, then we can approximate $c(x, t_{m-1})$ in (1) by

$$c(x, t_{m-1}) = e^{-r\Delta t} \int_a^b v(y, t_m) f(y|x) dy + \epsilon_1. \tag{4}$$

(The different error terms, ϵ_i, are discussed in Section 3.) As a second step, we replace the density function by its Fourier-cosine series expansion on $[a,b]$,

$$f(y|x) = {\sum_{k=0}^{\infty}}' A_k(x) \cos\left(k\pi \frac{y-a}{b-a}\right), \tag{5}$$

where $\sum'$ indicates that the first term in the summation is multiplied by 1/2. The series coefficients $\{A_k(x)\}_{k=0}^{\infty}$ are defined by

$$A_k(x) := \frac{2}{b-a}\int_a^b f(y|x)\cos\left(k\pi\frac{y-a}{b-a}\right)dy. \tag{6}$$

Interchanging the summation and integration operators yields

$$c(x,t_{m-1}) = \frac{1}{2}(b-a)e^{-r\Delta t}{\sum}'^{\infty}_{k=0} A_k(x)V_k(t_m) + \epsilon_1, \tag{7}$$

with $V_k(t_m)$ the Fourier-cosine series coefficients of $v(y,t_m)$ on $[a,b]$, i.e.

$$V_k(t_m) := \frac{2}{b-a}\int_a^b v(y,t_m)\cos\left(k\pi\frac{y-a}{b-a}\right)dy. \tag{8}$$

As a third step, we use the relation between $A_k(x)$ and the conditional characteristic function, $\phi(\omega;x)$, defined as

$$\phi(\omega;x) := \int_{\mathbb{R}} f(y|x)e^{i\omega y}dy. \tag{9}$$

Coefficients $A_k(x)$ can be written as

$$A_k(x) = \frac{2}{b-a}\mathrm{Re}\left\{e^{-ik\pi\frac{a}{b-a}}\int_a^b e^{i\frac{k\pi}{b-a}y}f(y|x)dy\right\}. \tag{10}$$

where $\mathrm{Re}\{\cdot\}$ denotes taking the real part. With (3), the finite integration in (10) can be approximated by

$$\int_a^b e^{i\frac{k\pi}{b-a}y}f(y|x)dy \approx \int_{\mathbb{R}} e^{i\frac{k\pi}{b-a}y}f(y|x)dy =: \phi\left(\frac{k\pi}{b-a};x\right).$$

As a result, $A_k(x)$ can be approximated by $F_k(x)$ with

$$F_k(x) := \frac{2}{b-a}\mathrm{Re}\left\{\phi\left(\frac{k\pi}{b-a};x\right)e^{-ik\pi\frac{a}{b-a}}\right\}. \tag{11}$$

Replacing $F_k(x)$ by $A_k(x)$ and then truncating the infinite series summation gives the COS formula for pricing European options for different underlying processes, which reads

$$\hat{c}(x,t_{m-1}) := e^{-r\Delta t}{\sum}'^{N-1}_{k=0}\mathrm{Re}\left\{\phi\left(\frac{k\pi}{b-a};x\right)e^{-ik\pi\frac{a}{b-a}}\right\}V_k(t_m). \tag{12}$$

Here the function $\hat{c}(x,t_{m-1})$ represents the approximation of the continuation value $c(x,t_{m-1})$. An error analysis justifying the different approximations for European options was presented in [22].

For exponential Lévy processes, formula (12) can be simplified to

$$\hat{c}(x,t_{m-1}) = e^{-r\Delta t}{\sum}'^{N-1}_{k=0}\mathrm{Re}\left\{\varphi_{levy}\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\right\}V_k(t_m), \tag{13}$$

where $\varphi_{levy}(\omega) := \phi_{levy}(\omega;0)$, see [22]. Using this, we can also approximate $v(x,t_0)$ in (2) by

$$\hat{v}(x,t_0) = e^{-r\Delta t}{\sum}'^{N-1}_{k=0}\mathrm{Re}\left\{\varphi_{levy}\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\right\}V_k(t_1), \tag{14}$$

provided that the series coefficients, $V_k(t_1)$, are known. We will show that the $V_k(t_m)$, $k=0,1,\cdots,N-1$, can be recovered from $V_j(t_{m+1})$, $j=0,1,\cdots,N-1$.

2.2 Series Coefficients of Option Values

The integral in the definition of $V_k(t_m)$ in (8) can be split into two parts when we know the *early-exercise point*, x_m^*, at time t_m, which is the point where the continuation value equals the payoff, $c(x_m^*, t_m) = g(x_m^*, t_m)$.

Since $\hat{c}(x, t_m)$ in (13) is an approximation of the whole function $c(x, t_m)$, and not only at grid points, we can simply use Newton's method to locate x_m^*. Note that, on each time lattice, there is at most one point which satisfies $\hat{c}(x, t_m) - g(x, t_m) = 0$. Therefore, we determine whether x_m^* lies in $[a, b]$ and, if not, set x_m^* equal to the nearest boundary point.

Once we have x_m^*, we can split the integral, which defines $V_k(t_m)$, into two parts: One on the interval $[a, x_m^*]$ and the second on $(x_m^*, b]$, i.e.

$$V_k(t_m) = \begin{cases} C_k(a, x_m^*, t_m) + G_k(x_m^*, b), & \text{for a call,} \\ G_k(a, x_m^*) + C_k(x_m^*, b, t_m), & \text{for a put,} \end{cases} \tag{15}$$

for $m = M-1, M-2, \cdots, 1$, and

$$V_k(t_M) = \begin{cases} G_k(0, b), \text{ for a call} \\ G_k(a, 0), \text{ for a put,} \end{cases} \tag{16}$$

whereby

$$G_k(x_1, x_2) := \frac{2}{b-a} \int_{x_1}^{x_2} g(x, t_m) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{17}$$

and

$$C_k(x_1, x_2, t_m) := \frac{2}{b-a} \int_{x_1}^{x_2} \hat{c}(x, t_m) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{18}$$

Theorem 2.1. *The $G_k(x_1, x_2)$ in (17) are known analytically and the $C_k(x_1, x_2, t_m)$ in (18) can be computed in $O(N \log_2(N))$ operations with the help of the Fast Fourier Transform (FFT).*

Proof. Let us first derive the analytical solution of $G_k(x_1, x_2)$. Since $g(x, t_m) \equiv \alpha K(1 - e^x)^+$, it follows, for a put, with $x_2 \le 0$, that

$$G_k(x_1, x_2) = \frac{2}{b-a} \int_{x_1}^{x_2} K(1 - e^x) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{19}$$

and for a call, with $x_1 \ge 0$, that

$$G_k(x_1, x_2) = \frac{2}{b-a} \int_{x_1}^{x_2} K(e^x - 1) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{20}$$

The fact that $x_m^* \le 0$ for put options and $x_m^* \ge 0$ for call options, $\forall t \in \mathcal{T}$, gives

$$G_k(x_1, x_2) = \frac{2}{b-a} \alpha K \left[\chi_k(x_1, x_2) - \psi_k(x_1, x_2)\right], \quad \alpha = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put,} \end{cases} \tag{21}$$

with

$$\chi_k(x_1, x_2) := \int_{x_1}^{x_2} e^x \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{22}$$

$$\psi_k(x_1, x_2) := \int_{x_1}^{x_2} \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{23}$$

These integrals admit the following analytical solutions:

$$
\begin{aligned}
\chi_k(x_1,x_2) &= \frac{1}{1+\left(\frac{k\pi}{b-a}\right)^2}\left[\cos\left(k\pi\frac{x_2-a}{b-a}\right)e^{x_2}-\cos\left(k\pi\frac{x_1-a}{b-a}\right)e^{x_1}\right.\\
&\quad\left.+\frac{k\pi}{b-a}\sin\left(k\pi\frac{x_2-a}{b-a}\right)e^{x_2}-\frac{k\pi}{b-a}\sin\left(k\pi\frac{x_1-a}{b-a}\right)e^{x_1}\right],\\
\psi_k(x_1,x_2) &= \begin{cases}\left[\sin\left(k\pi\frac{x_2-a}{b-a}\right)-\sin\left(k\pi\frac{x_1-a}{b-a}\right)\right]\frac{b-a}{k\pi} & k\neq 0,\\ (d-c) & k=0.\end{cases}
\end{aligned}
\tag{24}
$$

Next, we derive the formula for the coefficients $C_k(x_1,x_2,t_m)$. Notice that $c(x,t_m)$ in the definition of $C_k(x_1,x_2,t_m)$ in (18) has been replaced by approximation $\hat{c}(x,t_m)$, which yields

$$
C_k(x_1,x_2,t_m)=e^{-r\Delta t}\mathrm{Re}\left\{{\sum_{j=0}^{N-1}}'\varphi_{levy}\left(\frac{j\pi}{b-a}\right)V_j(t_{m+1})\cdot M_{k,j}(x_1,x_2)\right\},
\tag{25}
$$

where the coefficients $M_{k,j}(x_1,x_2)$ are given by

$$
M_{k,j}(x_1,x_2):=\frac{2}{b-a}\int_{x_1}^{x_2}e^{ij\pi\frac{x-a}{b-a}}\cos\left(k\pi\frac{x-a}{b-a}\right)dx,
\tag{26}
$$

with $i=\sqrt{-1}$ being the imaginary unit. With fundamental calculus, we can rewrite $M_{k,j}$ as

$$
M_{k,j}(x_1,x_2)=-\frac{i}{\pi}\left(M^c_{k,j}(x_1,x_2)+M^s_{k,j}(x_1,x_2)\right),
\tag{27}
$$

where

$$
M^c_{k,j}:=\begin{cases}\dfrac{(x_2-x_1)\pi i}{(b-a)} & k=j=0\\[2ex] \dfrac{\exp\left(i(j+k)\dfrac{(x_2-a)\pi}{b-a}\right)-\exp\left(i(j+k)\dfrac{(x_1-a)\pi}{b-a}\right)}{j+k} & \text{otherwise}\end{cases}
\tag{28}
$$

and

$$
M^s_{k,j}:=\begin{cases}\dfrac{(x_2-x_1)\pi i}{b-a} & k=j\\[2ex] \dfrac{\exp\left(i(j-k)\dfrac{(x_2-a)\pi}{b-a}\right)-\exp\left(i(j-k)\dfrac{(x_1-a)\pi}{b-a}\right)}{j-k} & k\neq j\end{cases}
\tag{29}
$$

In matrix-vector-product form, (25) reads

$$
\mathbf{C}(x_1,x_2,t_m)=\frac{e^{-r\Delta t}}{\pi}\mathrm{Im}\left\{(M_c+M_s)\mathbf{u}\right\},
\tag{30}
$$

where $\mathrm{Im}\{\cdot\}$ denotes taking the imaginary part, and

$$
\mathbf{u}:=\{u_j\}_{j=0}^{N-1},\quad u_j:=\varphi\left(\frac{j\pi}{b-a}\right)V_j(t_{m+1}),\ u_0=\frac{1}{2}\varphi\left(0\right)V_0(t_{m+1}).
\tag{31}
$$

Moreover, the matrices

$$M_c := \{M^c_{k,j}(x_1,x_2)\}_{k,j=0}^{N-1} \qquad \text{and} \qquad M_s := \{M^s_{k,j}(x_1,x_2)\}_{k,j=0}^{N-1}$$

have a special structure for which the FFT can be employed to compute (30) efficiently: Matrix M_c is a *Hankel* matrix,

$$M_c = \begin{bmatrix} m_0 & m_1 & m_2 & \cdots & m_{N-1} \\ m_1 & m_2 & \cdots & \cdots & m_N \\ \vdots & & & & \vdots \\ m_{N-2} & m_{N-1} & \cdots & & m_{2N-3} \\ m_{N-1} & \cdots & & m_{2N-3} & m_{2N-2} \end{bmatrix}_{N\times N} \tag{32}$$

and M_s is a *Toeplitz* matrix,

$$M_s = \begin{bmatrix} m_0 & m_1 & \cdots & m_{N-2} & m_{N-1} \\ m_{-1} & m_0 & m_1 & \cdots & m_{N-2} \\ \vdots & & \ddots & & \vdots \\ m_{2-N} & \cdots & m_{-1} & m_0 & m_1 \\ m_{1-N} & m_{2-N} & \cdots & m_{-1} & m_0 \end{bmatrix}_{N\times N} \tag{33}$$

with

$$m_j := \begin{cases} \dfrac{(x_2-x_1)\pi}{b-a} i & j=0 \\ \dfrac{\exp\left(ij\dfrac{(x_2-a)\pi}{b-a}\right) - \exp\left(ij\dfrac{(x_1-a)\pi}{b-a}\right)}{j} & j \neq 0 \end{cases} \tag{34}$$

Computation of $\mathbf{C}(x_1,x_2,t_m)$. For the computation of $\mathbf{C}(x_1,x_2,t_m)$ in (30), we require efficient algorithms for matrix-vector products, with a Toeplitz matrix, M_s, and a Hankel matrix, M_c. Due to the special structure of these matrices, we can rewrite these products into circular convolutions, that can be efficiently dealt with by the FFT algorithm. For Toeplitz matrices this is well-known, described in detail, for example, in [2]. The product $M_s\mathbf{u}$ is equal to the first N elements of $\mathbf{m}_s \circledast \mathbf{u}_s$ with the $2N$-vectors:

$$\mathbf{m}_s = [m_0, m_{-1}, m_{-2}, \cdots, m_{1-N}, 0, m_{N-1}, m_{N-2}, \cdots, m_1]^T,$$

and $\mathbf{u}_s = [u_0, u_1, \cdots, u_{N-1}, 0, \cdots, 0]^T$.

For the Hankel matrix this is less known, so we formulate it in the following result:

Result 2.1. *The product $M_c\mathbf{u}$ is equal to the first N elements of $\mathbf{m}_c \circledast \mathbf{u}_c$, in reversed order, with the $2N$-vectors: $\mathbf{m}_c = [m_{2N-1}, m_{2N-2}, \cdots, m_1, m_0]^T$ and $\mathbf{u}_c = [0, \cdots, 0, u_0, u_1, \cdots, u_{N-1}]^T$.*

For the efficient computation of $M_c\mathbf{u}$, we need to construct the following circulant matrix, M_u,

$$M_u = \begin{bmatrix} 0 & u_{N-1} & u_{N-2} & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & u_{N-1} & u_{N-2} & \cdots & \cdots & 0 \\ \vdots & & \ddots & & \ddots & & \vdots \\ 0 & \cdots & 0 & u_{N-1} & u_{N-2} & \cdots & u_0 \\ u_0 & 0 & \cdots & 0 & u_{N-1} & \cdots & u_1 \\ u_1 & u_0 & 0 & \cdots & 0 & \cdots & u_2 \\ \vdots & & \ddots & & & \ddots & \vdots \\ u_{N-2} & \cdots & u_0 & 0 & \cdots & 0 & u_{N-1} \\ u_{N-1} & u_{N-2} & \cdots & u_0 & 0 & \cdots & 0 \end{bmatrix}_{(2N)\times(2N)} . \tag{35}$$

Straightforward computation shows that the first N elements of the product of M_u and $\mathbf{m}_c$ equal $M_c\mathbf{u}$, in reversed order.

The FFT algorithm can be employed since the circular convolution of two vectors is equal to the inverse discrete Fourier transform ($\mathcal{D}^{-1}$) of the products of the forward DFTs, $\mathcal{D}$, i.e.,

$$\mathbf{x} \circledast \mathbf{y} = \mathcal{D}^{-1}\{\mathcal{D}(\mathbf{x}) \cdot \mathcal{D}(\mathbf{y})\}.$$

□

We can recover $V_k(t_1)$ recursively, backwards in time. Since the computation of $G_k(x_1, x_2)$ is linear in N, the overall complexity of this recovery procedure is dominated by the computation of $C_k(x_1, x_2, t_m)$, whose complexity is $N \log_2 N$ with the FFT. As a result, the overall computational complexity for pricing a Bermudan option with M exercise dates is $O((M-1)N\log_2 N)$, as the work needed for the final exercise is only $O(N)$.

2.3 The COS algorithm for Bermudan options

The pricing algorithm for Bermudan options is summarized into Algorithm 1:

Algorithm 1: Pricing Bermudan options with the COS method.

Initialization: For $k = 0, 1, \cdots, N-1$,

- $V_k(t_M) = G_k(0, b)$ for call options; $V_k(t_M) = G_k(a, 0)$ for put options;

Main Loop to Recover $V_k(t_m)$: For $m = M-1$ to 1,

- Determine early-exercise point x_m^* by Newton's method;
- Compute $V_k(t_m)$ from (15) (with the help of the FFT algorithm).

Final step: Reconstruct $v(x, t_0)$ by inserting $V_k(t_1)$ into (2).

To compute the Greeks, one only needs to modify the final step in Algorithm 1, from t_1 to t_0, as the Greeks can be approximated by

$$\hat{\Delta} = e^{-r\Delta t}\frac{2}{b-a}{\sum_{k=0}^{N-1}}{}' \mathrm{Re}\left\{\left\{\varphi\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\frac{ik\pi}{b-a}\right\}\right\}\frac{V_k(t_1)}{S_0} \tag{36}$$

and

$$\hat{\Gamma} = e^{-r\Delta t}\frac{2}{b-a}{\sum}'^{N-1}_{k=0}\mathrm{Re}\left\{\left\{\varphi\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\left[-\frac{ik\pi}{b-a}+\left(\frac{ik\pi}{b-a}\right)^2\right]\right\}\right\}\frac{V_k(t_1)}{S_0^2}. \tag{37}$$

The FFT algorithm is required five times for the computation of $\mathbf{C}(x_1, x_2, t_m)$, as detailed in the following algorithm.

ALGORITHM 2: Computation of $\mathbf{C}(x_1, x_2, t_m)$.

1. Compute $m_j(x_1, x_2)$ for $j = 0, 1, \cdots, N-1$ using (34).
2. Construct $\mathbf{m_s}(x_1, x_2)$ and $\mathbf{m_c}(x_1, x_2)$ using the properties of m_j's.
3. Compute $\mathbf{u}(t_m)$ using (31).
4. Construct $\mathbf{u_s}$ by padding N zeros to $\mathbf{u}(t_m)$.
5. $\mathbf{Msu}$ = the first N elements of $\mathcal{D}^{-1}\{\ \mathcal{D}(\mathbf{m_s})\cdot\mathcal{D}(\mathbf{u_s})\ \}$.
6. $\mathbf{Mcu}$ = reverse{ the first N elements of $\mathcal{D}^{-1}\{\ \mathcal{D}(\mathbf{m_c})\cdot\mathbf{sgn}\cdot\mathcal{D}(\mathbf{u_s})\ \}\}$.
7. $\mathbf{C}(x_1, x_2, t_m) = e^{-r\Delta t}\mathrm{Im}\left\{\mathbf{Msu} + \mathbf{Mcu}\right\}/\pi$.

Note that the operation $\mathcal{D}(\mathbf{u_s})$ is computed only once, and"reverse$\{\mathbf{x}\}$" denotes an $\mathbf{x}$-generated vector, whose elements are the same as those of $\mathbf{x}$ but sorted in reversed order.

Remark 2.1 (Efficient computation)**.** *It is worth mentioning that the computation of the exponentials takes significantly more computer clock cycles than additions or multiplications. One can however benefit from some special properties of the m_j's, like $m_{-j} = -\overline{m_j}$ and, for $j \neq 0$,*

$$m_{j+N} = \frac{\exp\left(iN\frac{(x_2-a)\pi}{b-a}\right)\cdot\exp\left(ij\frac{(x_2-a)\pi}{b-a}\right) - \exp\left(iN\frac{(x_1-a)\pi}{b-a}\right)\cdot\exp\left(ij\frac{(x_1-a)\pi}{b-a}\right)}{j+N}.$$

So, in order to construct $\mathbf{m}_s$ and $\mathbf{m}_c$, the factors $\exp\left(ij\frac{(x_2-a)\pi}{b-a}\right)$ and $\exp\left(ij\frac{(x_1-a)\pi}{b-a}\right)$, for $j = 0, 1\cdots, N-1$, should be computed only once.

Also, the DFT of $\mathbf{u}_c$ and of $\mathbf{u}_s$ need not be computed separately, as the shift property of DFTs gives $\mathcal{D}(\mathbf{u}_c) = \mathbf{sgn}\cdot\mathcal{D}(\mathbf{u}_s)$ with $\mathbf{sgn} = [1, -1, 1, -1, \cdots]^T$.

Remark 2.2 (Use of FFT algorithm)**.** *In the main loop of the CONV method from [32], the FFT algorithm is also called five times, and the length of the input vectors is halved compared to the COS method. Therefore, the CONV method would be approximately twice as fast, if we did not take the method's accuracy into account. However, for models characterized by density functions in* $\mathrm{C}^\infty[a, b]$*, the COS method exhibits an exponential convergence rate, which is superior to the second order convergence of the CONV method. For the same level of accuracy, the COS method is therefore significantly faster than the CONV method.*

2.4 Discretely-Monitored Barrier Options

Discretely-monitored "out" barrier options are options that cease to exist if the asset price hits a certain barrier level, H, at one of the pre-specified observation dates. If $H > S_0$, they are called "up-and-out" options, and "down-and-out" otherwise. The payoff for an up-and-out option reads

$$v(x,T) = (\max(\alpha(S_T - K), 0) - Rb)\mathbf{1}_{\{S_{t_i}<H\}} + Rb, \tag{38}$$

where $\alpha = 1$ for a call and $\alpha = -1$ for a put, Rb is a rebate, and $\mathbf{1}_A$ is the indicator function,

$$\mathbf{1}_A = \begin{cases} 1 & A \text{ is not empty,} \\ 0 & \text{otherwise.} \end{cases}$$

With the set of observation dates, $\mathcal{T} = \{t_1, \cdots, t_M\}$, $t_1 < \cdots < t_{M-1} < t_M = T$, the price of an up-and-out option, monitored M times, satisfies the following recursive formula

$$\begin{cases} c(x,t_{m-1}) &= e^{-r(t_m - t_{m-1})} \int_{\mathbb{R}} v(x,t_m) f(y|x) dy, \\ v(x,t_{m-1}) &= \begin{cases} e^{-r(T-t_{m-1})}Rb, & x \geq h, \\ c(x,t_{m-1}), & x < h, \end{cases} \end{cases} \tag{39}$$

where $h := \ln(H/K)$ and $m = M, M-1, \cdots, 2$.

Note that the recursive pricing formula (39) is very similar to that for the Bermudan options. What makes barrier pricing easier is that the root-searching algorithm is not needed as the barrier points are known in advance. Thus, similar to Bermudan options, discrete barrier options can be priced in two steps:

1. Recovery of the Fourier-cosine series coefficients of the option value at t_1,

2. The COS formula for European options given by (14).

Based on the derivation for Bermudan options, we have the following lemma:

Lemma 2.1 (Backward Induction for Discrete Barrier Options)**.** *By backward recursion we find the following solution for discretely monitored barrier options: For $m = M-1, M-2, \cdots, 1$,*

$$V_k(t_m) = C_k(a,h,t_m) + e^{-r(T-t_{m-1})}Rb\frac{2}{b-a}\psi_k(h,b) \tag{40}$$

with $C_k(x_1,x_2,t_m)$ and $\psi_k(x_1,x_2)$ given by (30) and (24), respectively. If $h < 0$, we have

$$V_k(t_M) = \begin{cases} 2Rb\psi_k(h,b)/(b-a) & \textit{for a call,} \\ G_k(a,h) + 2Rb\psi_k(h,b)/(b-a) & \textit{for a put.} \end{cases} \tag{41}$$

For $h \geq 0$, we find

$$V_k(t_M) = \begin{cases} G_k(0,h) + 2Rb\psi_k(h,b)/(b-a) & \textit{for a call,} \\ G_k(a,0) + 2Rb\psi_k(h,b)/(b-a) & \textit{for a put.} \end{cases} \tag{42}$$

A similar recursion formula for a down-and-out option can be derived easily.

Proof. The proof is straightforward, as it goes along the lines of the proof of Theorem 2.1. □

The computation of $\mathbf{C}(a, h, t_m)$ via (30) is less expensive than that of $\mathbf{C}(a, x_m^*, t_m)$ for Bermudan options, because h is known in advance, and consequently, $\psi_k(h, b)$ in (23), M_c and M_s in (30) are known before the recursion step. Therefore, the FFT technique is required only three times.

Barrier options with an "in" barrier can be priced as easily with the COS method. Alternatively, one could apply the barrier parity and symmetry results on "out" barrier options [42, 25].

We summarize the method by means of the following algorithm:

Algorithm 3: Pricing Discrete Barrier Options by the COS Method

Initialization:

- Compute $V_k(t_M)$ using (41) or (42) .
- For up-and-out: $x_1 = a$ and $x_2 = h$, and $c = h$ and $d = b$;
  For down-and-out: $x_1 = h$ and $x_2 = b$, and $c = a$ and $d = h$.
- Construct $\mathbf{m_s}(x_1, x_2)$ and $\mathbf{m_c}(x_1, x_2)$ using the properties of m_j's.
- $d_1 = \mathcal{D}\{\mathbf{m_s}(x_1, x_2)\}$, $d_2 = \mathbf{sgn} \cdot \mathcal{D}\{\mathbf{m_c}(x_1, x_2)\}$
- $\mathbf{G} = \frac{2}{b-a} Rb \left\{\psi_k(c, d)\right\}_{k=0}^{N-1}$.

Main Loop to Recover $\mathbf{V}(t_{m-1})$: For $m = M$ to 2,

1. Compute $\mathbf{u}(t_m)$ using Equation (31).
2. Construct $\mathbf{u_s}$ by padding N zeros to $\mathbf{u}(t_m)$.
3. $\mathbf{Msu}$ = the first N elements of $\mathcal{D}^{-1}\{\ d_1 \cdot \mathcal{D}(\mathbf{u_s})\ \}$.
4. $\mathbf{Mcu}$ = reverse$\{$ the first N elements of $\mathcal{D}^{-1}\{\ d_2 \cdot \mathcal{D}(\mathbf{u_s})\ \}\quad \}$.
5. $\mathbf{C}(t_{m-1}) = e^{-r\Delta t}/\pi \mathrm{Im}\left\{\mathbf{Msu} + \mathbf{Mcu}\right\}$.
6. $\mathbf{V}(t_{m-1}) = \mathbf{C}(t_{m-1}) + e^{-r(T-t_{m-1})}\mathbf{G}$

Finalization: Compute $v(t_0, x)$ according to (2); Or Greeks by (36) and (37).

3 Error Analysis

In [22], convergence and error analysis for European option pricing with the COS method were presented. Here we summarize the main conclusions. The generalization especially to barrier options (as we do not take the Newton step explicitly into account) is done in Subsection 3.2.

3.1 Convergence for European Options

In the derivation of the COS formula for European options, errors are introduced in three steps: truncation of the integration range of the risk-neutral valuation formula (4); substitution of the series coefficients of the density function by an approximation depending on the characteristic function (11); truncation of the infinite summation of the series (12). The insights in these errors in [22] were the following:

1. The integration range truncation error:

$$\epsilon_1 := \int_{\mathbb{R}\setminus[a,b]} v(y,T) f(y|x) dy. \tag{43}$$

   Naturally, the larger the truncation range, the smaller ϵ_1 gets.

2. The series truncation error,

$$\epsilon_2 := \frac{1}{2}(b-a)e^{-r\Delta t}\sum_{k=N}^{\infty} A_k(x)\cdot V_k, \tag{44}$$

   converges exponentially for probability density functions in the class $\mathrm{C}^{\infty}([a,b])$, i.e.

$$|\epsilon_2| < P\cdot \exp(-(N-1)\nu), \tag{45}$$

   where $\nu > 0$ is a constant and P is a term that varies less than exponentially with N. When the underlying density has a discontinuous derivative, the Fourier-cosine expansion converges *algebraically*, i.e.

$$|\epsilon_2| < \frac{\bar{P}}{(N-1)^{\beta-1}}, \tag{46}$$

   where $\bar{P}$ is a constant and $\beta \geq n \geq 1$ (and n is the algebraic index of convergence of the series coefficients).

3. The error of approximating $A_k(x)$

$$\epsilon_3 = e^{-r\Delta t}{\sum_{k=0}^{N-1}}{}' \mathrm{Re}\left\{\int_{\mathbb{R}\setminus[a,b]} e^{ik\pi\frac{y-a}{b-a}} f(y|x) dy\right\} V_k, \tag{47}$$

   can be bounded by:

$$|\epsilon_3| < |\epsilon_1| + Q\,|\epsilon_4|\,. \tag{48}$$

   Here, Q is some constant independent of N and

$$\epsilon_4 := \int_{\mathbb{R}\setminus[a,b]} f(y|x) dy = \epsilon_1.$$

   So, a large integration range reduces the size of both ϵ_1 and ϵ_3.

The numerical error of the COS method for European options, denoted by ϵ, can therefore be bounded by

$$|\epsilon| \leq 2\,|\epsilon_1| + |\epsilon_2| + |\epsilon_4|\,, \tag{49}$$

meaning that, with a properly chosen range of integration, component ϵ_2, i.e., the series truncation error, dominates.

3.2 Error Propagation in the Backward Induction

When the coefficients $V_k(t_1)$ are recovered recursively, backwards in time, the error, ϵ, may propagate through time. It is therefore necessary to analyze the method's stability through time.

Let us assume that $V_k(t_{m+2})$ is exact, implying that $\hat{c}(x, t_{m+1})$ obtained by the COS method contains error ϵ from (49). This error introduces, by substituting $\hat{c}(x, t_{m+1})$ in formula (18) for $C_k(x_1, x_2, t_{m+1})$, the error, $\varepsilon(k)$, defined by

$$\varepsilon(k) := \frac{2}{b-a}\int_{x_1}^{x_2} \epsilon \cos\left(k\pi\frac{x-a}{b-a}\right) dx = \frac{2\epsilon}{b-a}\psi_k(x_1, x_2).$$

Error $\varepsilon(k)$ can be interpreted as the product of error ϵ and the Fourier-cosine series coefficients of the following function $a(x)$:

$$a(x) = \begin{cases} 1, & x \in [x_1, x_2] \subset [a, b] \\ 0, & x \in \mathbb{R}\backslash[x_1, x_2]. \end{cases}$$

Let us denote the Fourier-cosine series coefficients of $a(x)$ by $\hat{A}_k$, then we have

$$\varepsilon(k) = \epsilon\hat{A}_k.$$

Error $\varepsilon(k)$ is present in the computation of $V_k(t_{m+1})$ in (13), which is therefore not exact. As a result, also an additional error component enters the computation of $\hat{c}(x, t_m)$, i.e.

$$\epsilon_5 := \epsilon\; e^{-r\Delta t}{\sum_{k=0}^{N-1}}' \mathrm{Re}\left\{\varphi\left(\frac{k\pi}{b-a}\right)e^{ik\pi\frac{x-a}{b-a}}\right\}\hat{A}_k. \tag{50}$$

Equation (50) can be viewed as an application of the COS method to a European option with $a(x)$ as the payoff function. We denote the exact value of this artificial option by $v_a(x)$, and find, based on the error analysis for European options, that

$$|\epsilon_5| = |\epsilon||v_a(x) + \epsilon|.$$

With the risk-neutral valuation formula, $v_a(x)$ can be bounded by

$$e^{r\Delta t}v_a(x) = \int_{\mathbb{R}} f(y|x)a(y)dy = \int_{x_1}^{x_2} f(y|x)dy \leq \int_{\mathbb{R}} f(y|x)dy = 1,$$

indicating that $v_a(x)$ is less than $e^{-r\Delta t}$. Putting the pieces together, we obtain the following bound:

$$|\epsilon_5| \leq |\epsilon|\left(e^{-r\Delta t} + |\epsilon|\right) \sim e^{-r\Delta t}\,|\epsilon|\,,$$

i.e., the local error remains of the same order, which is an indication for the COS method's stability.

3.3 Choice of Truncation Range

The insight from the error analysis in Section 3 is that the overall error consists of two parts: The series truncation error, which only depends on N and converges exponentially for processes whose density function belongs to $\mathrm{C}^{\infty}([a, b])$ (and algebraically otherwise), and the integration range error. We propose to use the following formula to define the range of integration in (3):

$$[a, b] := \left[(c_1 + x_0) - L\sqrt{c_2 + \sqrt{c_4}}, \quad (c_1 + x_0) + L\sqrt{c_2 + \sqrt{c_4}}\right], \tag{51}$$

where $x_0 := \ln(S_0/K)$ and L depends on the *user-defined tolerance level*, TOL, as given in (3). $c_1, \ldots, c_4$ are the cumulants, based on the characteristic function of the underlying process, and detailed in Appendix A.

Cumulant c_4 is included in (51), because, for short maturities, the density functions of many Lévy processes have sharp peaks and flat tails, and this behavior can be well captured by the inclusion of c_4.

Here, we analyze the relation between TOL and L in (51) via numerical experiments, aiming to determine one value of L for different exponential Lévy asset price processes . We present the observed error for different values of L in Figure 1. With N large, e.g. $N = 2^{14}$, the series truncation error is negligible and the integration range error, which has a direct relation to the user-defined TOL, dominates. The results in Figure 1 can therefore be used as a guidance for setting parameter L, given a tolerance TOL. In the figure, and throughout this paper, BS denotes the Black-Scholes model (Geometric Brownian Motion), VG stands for Variance Gamma model [33], CGMY denotes the model from [12], NIG is short for the Normal Inverse Gaussian Lévy process [5], Merton denotes the jump-diffusion model developed in [34], and Kou is the jump-diffusion model from [30]. We see in Figure 1 that the integration range

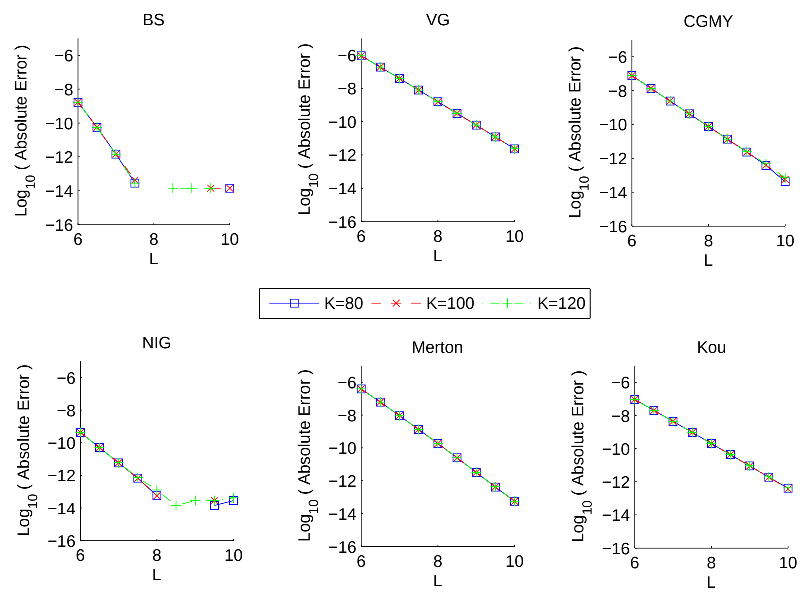


Figure 1: L versus the logarithm of the absolute errors for pricing calls by the COS method with $N = 2^{14}$, $T = 1$ year and three different strike prices.

error decreases exponentially with L. $L = 8$ seems an appropriate choice for all these Lévy processes. This value is used in all numerical experiments to follow.

Via experiments, we found that formula (51), together with a proper choice of L, defines an appropriate truncation range for any maturity time longer than 0.1 years. For even shorter maturities, one can use a larger value of L.

4 Numerical Results

We will show the method's impressive convergence by pricing Bermudan, American and discretely-monitored barrier options. In the following, we present numerical results for the BS, CGMY and NIG models. Extensive tests (not given here) have demonstrated that the COS method also shows excellent performance under other Lévy processes. The characteristic functions as well as the cumulants for several exponential Lévy asset price processes are listed in Appendix A.

The computer used has an Intel Pentium 4 CPU, 2.80GHz with cache size 1024 KB; The code is written in Matlab 7.4. The CPU times for all experiments to follow are averaged over 100 repeated tests.

In order to observe the exponential error convergence, we define a ratio,

$$\text{ratio} = \frac{\ln\left(\left|err(2^{d+1})\right|\right)}{\ln\left(\left|err(2^{d})\right|\right)}, \quad d \in \mathbb{Z}^+, \tag{52}$$

where $err(2^d)$ denotes the error, between reference solution and approximation obtained with $N = 2^d$. If $err(N) = C_1 \exp(-P_1 N)$ with C_1 and P_1 not depending on N, this ratio should be equal to 2; If the error convergence is algebraic, i.e. $err(N) = C_2 N^{-P_2}$ with C_2 and P_2 independent of N, this ratio should equal $(d+1)/d$.

Next to the series and the integration range truncation error, another error for Bermudan options comes from the stopping criterion of the root-searching algorithm, i.e., Newton's method. With an initial guess $x^*_{m+1} = x^*_m$, $m = M-2, \ldots, 2$ (and $x^*_{M-1} = 0$), this error becomes sufficiently small, of $O(10^{-7})$ by 4 Newton steps and of $O(10^{-10})$ by 5 steps. In the experiments to follow, we use 5 steps.

4.1 Bermudan and American Options

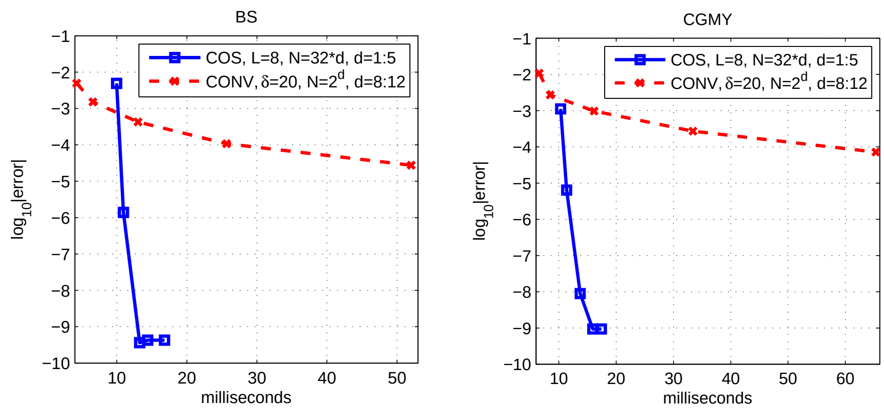
Here we price Bermudan put options with 10 exercise dates. Test parameters for two test cases are given in Table 1. These parameters are related to the characteristic functions presented in Table 8 and the cumulants from Table 9.

Table 1: Test parameters for pricing Bermudan options

Test No.	Model	S_0	K	T	r	σ	Other Parameters
1	BS	100	110	1	0.1	0.2	–
2	CGMY	100	80	1	0.1	0	$C=1, G=5, M=5, Y=1.5$

The CPU times are reported in milli-seconds, and all reference values are obtained by another method, i.e., the CONV method from [32], setting $N = 2^{20}$.

The first test is for the classical BS model with as the reference value 10.479520123. In Figure 2a it is shown that a highly accurate solution is obtained in less than 20 milli-seconds with exponential convergence (the log-error



(a) BS (Test No. 1) (b) CGMY with $Y = 1.5$ (Test No. 2)

Figure 2: Error versus CPU time for pricing Bermudan put options under (a) BS and (b) CGMY model, comparing the COS and the CONV method.

plot shows a straight line). Compared to the quadrature-rule based CONV method, which exhibits a second-order convergence, we see a significant improvement in the CPU time.

As the second test, we consider a Lévy process of infinite activity, i.e., the CGMY model with $Y > 1$ (Test 2 in Table 1). For this set of CGMY parameters it is now well-known that PIDE-based methods have convergence difficulties [2, 43]. The reference value is found to be $28.829781986\ldots$. The performance of the COS method for this test, shown in Figure 2b is highly efficient. Again, in less than 20 milli-seconds, the solution is accurate to 9 digits, compared to the reference value. Also here, we observe the exponential error convergence of the COS method.

Remark 4.1 (VG and Algebraic convergence)**.** *In [22] it was shown that for certain sets of parameters the Variance Gamma (VG) process gives rise to a probability density function which is not in* $\mathrm{C}^\infty(\mathbb{R})$*, and thus option pricing under VG with these parameter sets exhibits only an algebraic convergence. This is observed for contracts with* $T < \nu$*, where* ν *denotes the variance of the VG model, see the characteristic function in Appendix A.*

When dealing with Bermudan options this also implies that we will encounter algebraic convergence when the time between two exercise dates, $\Delta t < \nu$*.*

The prices of American options can be obtained by applying a *Richardson extrapolation* on prices of a few Bermudan options with small M [15], as demonstrated, for example, in [32]. Let $v(M)$ denote the value of a Bermudan option with M early exercise dates. We will use the following 4-point Richardson extrapolation scheme,

$$v_{AM}(d) = \frac{1}{21}\left(64v(2^{d+3}) - 56v(2^{d+2}) + 14v(2^{d+1}) - v(2^d)\right), \tag{53}$$

where $v_{AM}(d)$ denotes the approximated value of the American option.

Now we price an American option using (53) with the 4-point Richardson extrapolation on Bermudan puts and vary the number of exercise dates. The parameters, presented in Table 2, are taken from [1] and the reference value given was $V(0) = 0.112152$. We deal with the pure Lévy CGMY jump model ($\sigma = 0$) and no dividend payment ($q = 0$).

Table 2: Parameters for American put options under the CGMY model

Test No.	S_0	K	T	r	Other Parameters
3	1	1	1	0.1	$C = 1, G = 5, M = 5, Y = 0.5$

We compare the results of the COS method with those obtained by the CONV method using the same extrapolation. For the COS method, $N = 512$ and the number of Newton steps is 5; For the CONV method $N = 4096$ to reach a very similar accuracy. The accuracy of the American prices then mainly depends on parameter d in the extrapolation (53). Results are summarized in Table 3. We can see that high values of d give highly accurate results. The COS method in combination with Richardson extrapolation gives, however, a very satisfactory accuracy within 75 milli-seconds.

Table 3: Errors and CPU times for pricing American puts under CGMY model, Test No. 3

d in Eq. (53)	COS		CONV	
	error	time (milli-sec.)	error	time (milli-sec.)
0	4.41e-05	71.41	4.37e-05	134.4
1	7.69e-06	109.2	7.01e-06	198.0
2	9.23e-07	219.3	1.05e-06	336.7
3	3.04e-07	438.9	1.29e-07	610.9

4.2 Barrier Options

Now we price monthly-monitored ($M = 12$) up-and-out call and put options, (UOC) and (UOP), down-and-out call and put options, (DOC) and (DOP), by the COS method. The test parameters are in Table 4, again related to the characteristic functions in Table 8. We solve the same problems as in [23] with the barrier level, $H = 120$ for the up-and-out and $H = 80$ for the down-and-out options.

Table 4: Test parameters for pricing barrier options

Test No.	Model	S_0	K	T	r	q	Other Parameters
4	CGMY	100	100	1	0.05	0.02	$C = 4, G = 50, M = 60, Y = 0.7$
5	NIG	100	100	1	0.05	0.02	$\alpha = 15, \beta = -5, \delta = 0.5$

The numerical results under the CGMY model (Test 4) are presented in Table 5. The CPU times are again measured in milli-seconds, and the reference values are obtained by the CONV method [32], with $N = 2^{15}$. Note that “ratio”, as presented in the table, is different from the commonly used ratio defining the rate of convergence. In (52), it is the ratio of the *logarithm* of two consecutive errors. This ratio should be equal to two in the case of exponential convergence.

As expected, the COS method is more efficient for discrete barrier options than for Bermudan options, because the barrier levels are known in advance.

Exponential error convergence is observed, as the ratios (52) are around 2, in less than 5 milli-seconds with the results accurate up to 8 decimal places.

Table 5: Errors and CPU times for pricing monthly-monitored barrier options under the CGMY model (Test No. 4)

Option Type	Ref. Val.	N	time (milli-sec.)	error	ratio
DOP	2.339381026	2^4	2.8	2.23e-1	–
		2^5	2.7	1.98e-2	2.6
		2^6	3.4	3.23e-4	2.0
		2^7	4.6	7.20e-9	2.3
DOC	9.155070561	2^4	2.7	5.06e-2	–
		2^5	2.9	5.67e-3	1.7
		2^6	3.3	1.99e-4	1.6
		2^7	4.7	5.55e-9	2.2
UOP	6.195603554	2^4	3.0	5.58e-2	–
		2^5	2.9	8.98e-3	1.6
		2^6	3.6	1.96e-4	1.8
		2^7	4.8	2.23e-8	2.1
UOC	1.814827593	2^4	2.8	3.38e-1	–
		2^5	2.8	1.24e-2	4.0
		2^6	3.5	3.45e-6	2.9
		2^7	4.7	1.93e-8	1.4

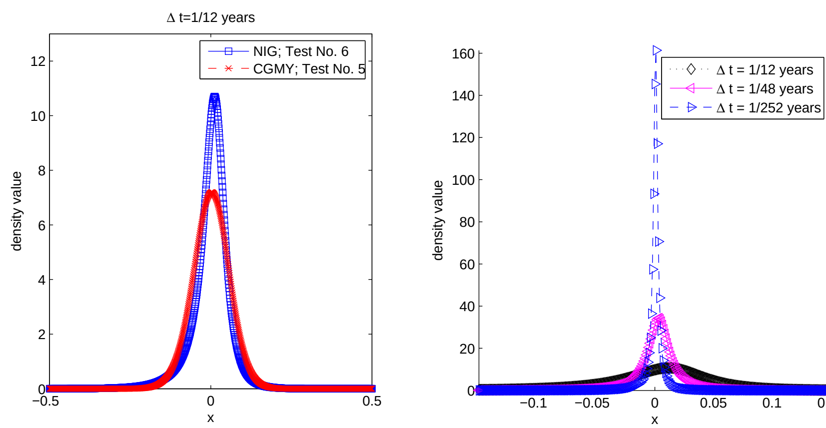
Next, we focus on the NIG model (Test 5) and repeat the barrier option tests in Table 6. To reach the same level of accuracy as for CGMY, we need a slightly larger value of N under the NIG model. This is because the NIG density function is more peaked, with the parameters from Table 4, as shown in Figure 3a. Consequently, one typically requires some more terms in the series expansion to reconstruct the density function from its Fourier-cosine series expansion. Nevertheless, the performance of the COS method is still excellent: In less than ten milli-seconds, the accuracy is up to the 7-th decimal place.

Note that, the smaller the value of Δt, the larger the value of N is required to reach the same level of accuracy. This is because many Lévy processes have highly peaked density functions for very small Δt. An example is presented in Figure 3b, where the recovered density functions of the NIG model for monthly-, weekly- and daily-monitored barrier options are plotted. We can see that for $\Delta t = 1/252$ the density is highly peaked, compared to $\Delta t = 1/12$. Nevertheless, as long as the density function is in $C^\infty(\mathbb{R})$, the error convergence rate is exponential.

We now price *daily-monitored* DOP and DOC options under the NIG model with the parameters from Test 5 in Table 4. The reference values are taken from [23]. Our results with the COS method are summarized in Table 7. We observe

Table 6: Errors and CPU times for pricing monthly-monitored barrier options under the NIG model (Test No. 5)

Option Type	Ref. Val.	N	time (milli-sec.)	error	ratio
DOP	2.139931117	2^6	3.1	4.25e-2	–
		2^7	3.7	1.28e-3	2.1
		2^8	5.4	4.65e-5	1.5
		2^9	8.4	1.39e-7	1.6
		2^{10}	14.7	1.38e-12	1.7
DOC	8.983106036	2^6	3.1	1.26e-2	–
		2^7	3.7	1.09e-3	1.6
		2^8	5.3	3.99e-5	1.5
		2^9	8.3	9.47e-8	1.6
		2^{10}	14.8	5.61e-13	1.7
UOP	5.995341168	2^6	3.4	4.84e-3	–
		2^7	3.7	1.14e-3	1.3
		2^8	5.3	7.50e-5	1.4
		2^9	8.3	1.52e-7	1.7
		2^{10}	14.7	1.24e-12	1.7
UOC	2.277861597	2^6	3.1	3.83e-2	–
		2^7	3.7	1.10e-3	2.1
		2^8	5.5	8.67e-5	1.4
		2^9	8.6	7.98e-8	1.7
		2^{10}	15.1	7.38e-13	1.7



(a) NIG and CGMY with $\Delta t = \frac{1}{12}$ (b) NIG with $\Delta t = \frac{1}{12}, \frac{1}{48}$, and $\frac{1}{252}$

Figure 3: The recovered density functions for (a) the NIG and the CGMY models and *monthly-monitored* barrier options and (b) the NIG model for *monthly-, weekly- and daily-monitored* barrier options.

that, as expected, the convergence rate of the COS method is exponential, but the values of N are somewhat larger than in the previous numerical experiments.

The almost linear computational complexity of the method can clearly be seen in this table.

Table 7: Errors and CPU times for pricing daily-monitored ($M = 252$) barrier options under the NIG model (Test 5).

Option Type	Ref. Val.	N	time (milli-sec.)	error	ratio
DOP	1.88148753	2^9	130	1.25e-2	–
		2^{10}	230	2.20e-3	1.4
		2^{11}	460	1.32e-4	1.5
		2^{12}	1170	1.98e-6	1.5
		2^{13}	2560	4.70e-8	1.3
DOC	8.96705248	2^9	140	3.67e-4	–
		2^{10}	230	9.18e-5	1.2
		2^{11}	460	3.14e-5	1.1
		2^{12}	950	2.00e-6	1.3
		2^{13}	2430	5.73e-9	1.4

For results accurate up to the 4th digit, the COS method needs about 0.2 seconds for the daily-monitored DOP as well as for the DOC.

Remark 4.2 (Comparison to Hilbert transform method)**.** *The complexity of the COS method is $O((M-1)N\log_2(N))$, as the length of the induction loop (in which the FFT is employed) is $M-1$, and the final step uses N operations. Additionally, its error convergence is exponential for models with density function in the class $\mathrm{C}^\infty([a,b])$. By considering both complexity and error convergence, the COS method is as efficient as the Hilbert transform method in [23]. The experiments above show that the COS method is as fast in terms of CPU time (although we have a slower CPU and the code is written in Matlab). That method cannot be used to price Bermudan options, as the information of the early-exercise points is not known in advance. Moreover, the COS method uses more-or-less the same CPU time for different types of barrier options, which is not the case in [23].*

5 Conclusions and Discussion

In this paper, we have generalized the COS option pricing method, based on Fourier-cosine expansions, to Bermudan and discretely-monitored barrier options. The method can be used whenever the characteristic function of the underlying price process is available (i.e., for regular affine diffusion processes and, in particular, for exponential Lévy processes).

The main insights in the paper are that the COS formula for European options from [22] can be used for pricing Bermudan and barrier options, if the series coefficients of the option values at the first early-exercise (or monitoring) date are known. These coefficients can be recursively recovered from those of the payoff function. The computational complexity is $O((M-1)N\log_2 N)$, for Bermudan and barrier options with M exercise, or monitoring, dates. The COS method exhibits an exponential convergence in N for density functions in

$C^\infty[a,b]$ and an impressive computational speed. With a small N, it typically produces highly accurate results. For example, with $N = 128$, results are accurate up to the 8th digit in less than 20 milli-seconds for 10-time exercisable Bermudan options and less than 10 milli-seconds for monthly-monitored barrier options. We expect a further factor of three gain is computational speed when replacing the Matlab implementation by an implementation in C.

However, the smaller the time interval between two consecutive dates, the more peaked the underlying density function, and thus larger values of N are required for a similar accuracy. Nonetheless, for problems with small time intervals, like daily-monitored barrier options, the COS method shows a similar performance as the Hilbert transform based method [23].

Compared to the CONV method [32], which is one of the fast methods for Bermudan options, the COS method converges significantly faster to the same level of accuracy. Pricing American options can be done by a Richardson extrapolation method on Bermudan options with a varying number of exercise dates.

References

[1] Almendral A. and Oosterlee C.W., Accurate evaluation of European and American options under the CGMY process., *SIAM J. Sci. Comput.* 29: 93-117, 2007.

[2] Almendral A. and Oosterlee C. W., On American options under the Variance Gamma process, *Appl. Math. Finance,* 14(2):131-152, 2007.

[3] Andricopoulos A.D., Widdicks M., Duck P.W. and Newton D.P., Universal option valuation using quadrature methods, *J. Fin. Economics,* 67: 447-471, 2003.

[4] Andricopoulos A.D., Widdicks M., Duck P.W. and Newton D.P., Extending quadrature methods to value multi-asset and complex path dependent options, *J. Fin. Economics,* 2006.

[5] Barndorff-Nielsen O. E. Normal inverse Gaussian distributions and stochastic volatility modelling, *Scand. J. Statist.*, 24:1-13, 1997.

[6] Bender C.M. and Orszag S.A., *Advanced mathematical methods for scientists and engineers.* McGraw-Hill, New York, 1978.

[7] Boyarchenko, S.I. and Levendorskiǐ, S.Z., *Non-Gaussian Merton-Black-Scholes theory*, Vol. 9 Advanced Series on Statist. Science & Appl. Probability., World Scient. Publ. Co. Inc., River Edge, NJ, 2002.

[8] Boyd J.P., *Chebyshev & Fourier spectral methods.* Springer-Verlag, Berlin, 1989.

[9] Broadie M. and Yamamoto Y., Application of the fast Gauss transform to option pricing, *Management Science,* 49(8): 1071-1088, 2003.

[10] Broadie M. and Yamamoto Y., A double-exponential fast Gauss transform for pricing discrete path-dependent options, *Operations Research,* 53(5): 764-779, 2005.

[11] Cariboni J. and Schoutens W., Pricing credit default swaps under Lévy models. *J. Comp. finance,* 10(4) 71-91, 2008.

[12] Carr P.P., Geman H., Madan D.B. and Yor M., The fine structure of asset returns: An empirical investigation. *J. of Business,* 75: 305-332, 2002.

[13] Carr P.P. and Madan D.B., Option valuation using the Fast Fourier Transform. *J. Comp. Finance,* 2:61-73, 1999.

[14] Chourdakis K., Option pricing using the Fractional FFT. *J. Comp. Finance* 8(2), 2004.

[15] Chang C-C, Chung S-L and Stapleton R.C., Richardson extrapolation technique for pricing American-style options. *J. Futures Markets,* 27(8): 791-817, 2007.

[16] Cont R. and Tankov P., *Financial modelling with jump processes*, Chapman and Hall, Boca Raton, FL, 2004.

[17] Dempster M.A.H. and Hong S.S.G., *Spread option valuation and the Fast Fourier transform.* Techn. Rep. WP 26/2000, the Judge Inst. Manag. Studies, Univ. Cambridge, 2000.

[18] Duffie D., Pan J. and Singleton K.J., Transform analysis and asset pricing for affine jump-diffusions. *Econometrica,* 68: 1343–1376, 2000.

[19] Duffie D., Filipovic D. and Schachermayer W., Affine processes and applications in finance. *Ann. of Appl. Probab.,* 13(3): 984-1053, 2003.

[20] Evans G.A. and Webster J.R., A comparison of some methods for the evaluation of highly oscillatory integrals. *J. of Comp. Applied Math.* 112: 55-69, 1999.

[21] Eydeland A., A fast algorithm for computing integrals in function spaces: financial applications. *Computational Economics* 7(4): 277-285, 1994.

[22] Fang, F. and Oosterlee, C.W. A novel option pricing method based on Fourier-cosine series expansions, submitted, 2008, see `http://ta.twi.tudelft.nl/mf/users/oosterle/oosterlee/COS.pdf`.

[23] Feng L. and Linetsky V., Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: a fast Hilbert transform approach, To appear in *Mathematical Finance*, 2008.

[24] Haug E.G., *The complete guide to option pricing formulas.* McGraw-Hill, 1998.

[25] Haug E.G., *Barrier put-call transformations*, see `http://www.smartquant.com/references/OptionPricing/option27.pdf`

[26] Heston S., A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financ. Studies,* 6: 327-343, 1993.

[27] Hirsa A. and Madan D.B., Pricing American options under Variance Gamma. *J. Comp. Finance,* 7, 2004.

[28] Hull J.C. *Options, futures and other derivatives.* Prentice Hall. 4th ed., 2000.

[29] Jackson K., Jaimungal S. and Surkov V., *Option pricing with regime switching Lévy processes using Fourier space time-stepping.* Proc. 4th IASTED Intern. Conf. Financial Engin. Applic., 92-97, 2007.

[30] Kou S. G., A jump diffusion model for option pricing. *Management Science,* 48(8): 1086-1101, 2002.

[31] Lewis A. *A simple option formula for general jump-diffusion and other exponential Lévy processes.* SSRN working paper, 2001. Available at: `http//ssrn.com/abstract=282110`.

[32] Lord R., Fang F., Bervoets F. and Oosterlee C.W., A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes. *SIAM J. Sci. Comput.,* 30: 1678-1705, 2008.

[33] Madan D. B., Carr P. R. and Chang E. C. The Variance Gamma process and option pricing. *European Finance Review,* 2: 79-105, 1998.

[34] Merton R. Option pricing when underlying stock returns are discontinuous, *J. Financial Economics,* 3: 125-144, 1976.

[35] O'Sullivan C., *Path dependent option pricing under Lévy processes* EFA 2005 Moscow Meetings Paper, Available at SSRN: `http://ssrn.com/abstract=673424`, 2005.

[36] Piessens R. and Poleunis F., A numerical method for the integration of oscillatory functions. *BIT,* 11: 317-327, 1971.

[37] Raible S., *Lévy processes in finance: Theory, numerics and empirical facts.* PhD Thesis, Inst. für Math. Stochastik, Albert-Ludwigs-Univ. Freiburg, 2000.

[38] Sato, K-I., *Basic results on Lévy processes, In:* Lévy processes, 3–37, Birkhäuser Boston, Boston MA, 2001.

[39] Schoutens, W., *Lévy processes in finance: Pricing financial derivatives.* Wiley, 2003.

[40] Singleton K.J. and Umantsev L., Pricing coupon-bond options and swaptions in affine term structure models. *Math. Finance*, 12(4): 427-446, 2002.

[41] Surkov V., *Parallel option pricing with Fourier space time-stepping method on graphics processing units.* Preprint Univ. of Toronto, 2007. See: `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1020207`

[42] Taleb N., *Dynamic Hedging.* John Wiley & Sons, New York, 2002.

[43] Wang I., Wan J.W. and Forsyth P., Robust numerical valuation of European and American options under the CGMY process. *J. Comp. Finance,* 10(4): 31-70, 2007.

[44] Wilmott P., *Derivatives: The theory and practice of financial engineering.* Wiley Frontiers in Finance Series, 1998.

A Characteristic Functions and Cumulants

The COS method requires from the underlying process the characteristic function to be known. The method fits therefore well to exponential Lévy models, whose characteristic functions are available in closed-form. The motivation behind using general Lévy processes for the underlying is the fact that the Black-Scholes model is not able to reproduce the volatility skew or smile present in most financial markets, whereas it has been shown that several exponential Lévy models can, at least to some extent.

In exponential Lévy models the asset price is modeled as an exponential function of a Lévy process $L(t)$:

$$S(t) = S(0)\exp(L(t)). \tag{54}$$

A process $L(t)$, with $L(0) = 0$, is a Lévy process if:

1 it has independent increments;

2 it has stationary increments;

3 it is stochastically continuous, i.e., for any $t \geq 0$ and $\epsilon > 0$ we have

$$\lim_{s \to t} \mathbb{P}(|L(t) - L(s)| > \epsilon) = 0. \tag{55}$$

Each Lévy process can be characterized by a triplet (μ, σ, ζ) with $\mu \in \mathbb{R}, \sigma \geq 0$ and ζ a measure satisfying $\zeta(0) = 0$ and

$$\int_{\mathbb{R}} \min(1, |x|^2)\zeta(dx) < \infty. \tag{56}$$

In terms of this triplet the characteristic function of the Lévy process is available in closed form, due to the celebrated Lévy-Khinchine formula. We recall the formulae for the characteristic function for several exponential Lévy processes in Table 8. For more background information on these processes we point you to [16, 39] for the usage of Lévy processes in a financial context and to [38] for a detailed analysis of Lévy processes in general. With respect to the paqrameters for the various processes in Table 8 we also basically follow the books [16, 39].

Given the characteristic functions, the cumulants, defined in [16], can be computed via

$$c_n(X) = \frac{1}{i^n} \frac{\partial^n (t\Psi(\xi))}{\partial \xi^n}\bigg|_{\xi=0},$$

where $t\Psi(\xi)$ is the exponent of the characteristic function $\varphi(\xi, t)$, i.e.

$$\varphi(\xi, t) = e^{t\Psi(\xi)}, \quad t \geq 0.$$

The formulae for the cumulants are summarized in Table 9. They have been confirmed with the help of Mathematica.

Table 8: Characteristic functions of $\ln(S_t/K)$ for various models.

BS	$\varphi(\xi,t) = \exp\left(i\xi\mu t - \frac{1}{2}\sigma^2\xi^2 t\right)$
NIG	$\varphi(\xi,t) = \exp\left(i\xi\mu t - \frac{1}{2}\sigma^2\xi^2 t\right)\phi_{NIG}(\xi,t;\alpha,\beta,\delta)$ $\phi_{NIG}(\xi,t;\alpha,\beta,\delta) = \exp\left[\delta t\left(\sqrt{\alpha^2-\beta^2} - \sqrt{\alpha^2-(\beta+i\xi)^2}\right)\right]$
Kou	$\varphi(\xi,t) = \exp\left(i\xi\mu t - \frac{1}{2}\sigma^2\xi^2 t\right)\phi_{Kou}(\xi,t;\lambda,p,\eta_1,\eta_2)$ $\phi_{Kou}(\xi,t;\lambda,p,\eta_1,\eta_2) = \exp\left[\lambda t\left(\frac{p\eta_1}{\eta_1-i\xi} - \frac{(1-p)\eta_2}{\eta_2+i\xi} - 1\right)\right]$
Merton	$\varphi(\xi,t) = \exp\left(i\xi\mu t - \frac{1}{2}\sigma^2\xi^2 t\right)\phi_{Merton}(\xi,t;\lambda,\bar{\mu},\bar{\sigma})$ $\phi_{Merton}(\xi,t) = \exp\left[\lambda t\left(\exp(i\bar{\mu}\xi - \frac{1}{2}\bar{\sigma}^2\xi^2) - 1\right)\right]$
VG	$\varphi(\xi,t) = \exp\left(i\xi\mu t\right)\phi_{VG}(\xi,t;\sigma,\nu,\theta)$ $\phi_{VG}(\xi,t;\sigma,\nu,\theta) = (1 - i\xi\theta\nu + \frac{1}{2}\sigma^2\nu\xi^2)^{-t/\nu}$
CGMY	$\varphi_{\ln(S_t/K)}(\xi,t;x) = \exp\left(i\xi\mu t - \frac{1}{2}\sigma^2\xi^2 t\right)\phi_{CGMY}(\xi,t;C,G,M,Y)$ $\phi_{CGMY}(\xi,t;C,G,M,Y) = \exp(Ct\Gamma(-Y)[(M-i\xi)^Y - M^Y + (G+i\xi)^Y - G^Y])$

Table 9: Cumulants of $\ln(S_t/K)$ for various models.

BS	$c_1 = (\mu - \frac{1}{2}\sigma^2)t, \quad c_2 = \sigma^2 t, \quad c_4 = 0$	
NIG	$c_1 = (\mu - \frac{1}{2}\sigma^2 + w)t + \delta t\beta/\sqrt{\alpha^2-\beta^2}$ $c_2 = \delta t\alpha^2(\alpha^2-\beta^2)^{-3/2}$ $c_4 = 3\delta t\alpha^2(\alpha^2+4\beta^2)(\alpha^2-\beta^2)^{-7/2}$ $w = -\delta(\sqrt{\alpha^2-\beta^2} - \sqrt{\alpha^2-(\beta+1)^2})$	
Kou	$c_1 = t\left(\mu + \frac{\lambda p}{\eta_1} + \frac{\lambda(1-p)}{\eta_2}\right)$ $c_4 = 24t\lambda\left(\frac{p}{\eta_1^4} + \frac{1-p}{\eta_2^4}\right)$	$c_2 = t\left(\sigma^2 + 2\frac{\lambda p}{\eta_1^2} + 2\frac{\lambda(1-p)}{\eta_2^2}\right)$ $w = \lambda\left(\frac{p}{\eta_1+1} - \frac{1-p}{\eta_2-1}\right)$
Merton	$c_1 = t(\mu + \lambda\bar{\mu})$ $c_4 = t\lambda\left(\bar{\mu}^4 + 6\bar{\sigma}^2\bar{\mu}^2 + 3\bar{\sigma}^4\lambda\right)$	$c_2 = t\left(\sigma^2 + \lambda\bar{\mu}^2 + \bar{\sigma}^2\lambda\right)$
VG	$c_1 = (\mu+\theta)t$ $c_4 = 3(\sigma^4\nu + 2\theta^4\nu^3 + 4\sigma^2\theta^2\nu^2)t$	$c_2 = (\sigma^2 + \nu\theta^2)t$ $w = \frac{1}{\nu}\ln(1 - \theta\nu - \sigma^2\nu/2)$
CGMY	$c_1 = \mu t + Ct\Gamma(1-Y)\left(M^{Y-1} - G^{Y-1}\right)$ $c_2 = \sigma^2 t + Ct\Gamma(2-Y)\left(M^{Y-2} + G^{Y-2}\right)$ $c_4 = Ct\Gamma(4-Y)\left(M^{Y-4} + G^{Y-4}\right)$ $w = -C\Gamma(-Y)[(M-1)^Y - M^Y + (G+1)^Y - G^Y]$	

where w is the drift correction term that satisfies $\exp(-wt) = \varphi(-i,t)$.